

Функції

# Програмні модулі мови c

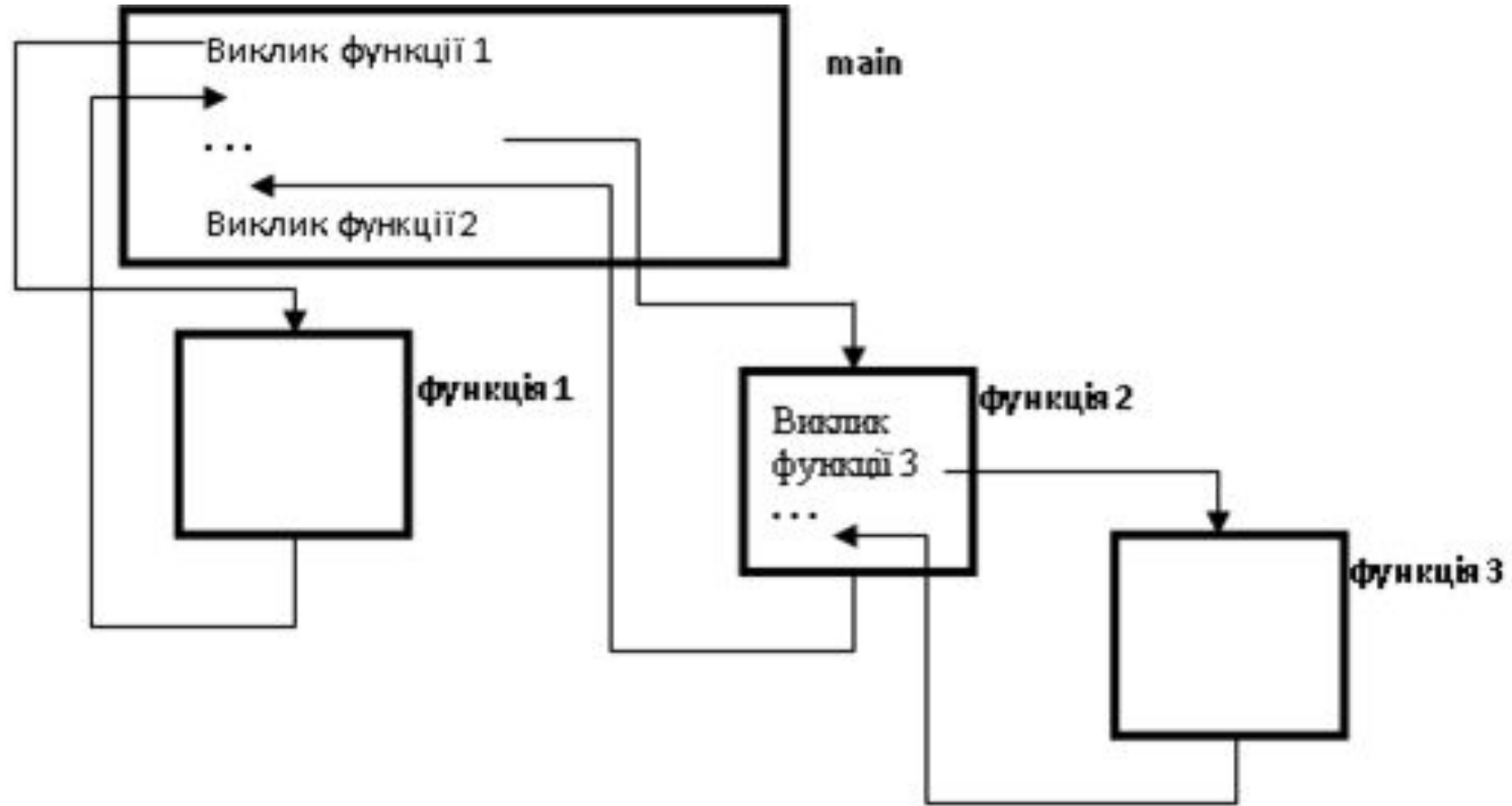
- Найкращим засобом розробки програми і підтримки великих програм є конструювання програми у вигляді невеликих окремих частин – модулів.
- Модулі мови C називаються **функціями**. Програми на C розроблюються, як правило, шляхом об'єднання нових функцій, які розроблені програмістом, із функціями, які поставляються у складі системної бібліотеки мови C.
- Стандартна бібліотека мови C надає велику сукупність функцій для виконання загальних математичних обчислень, обробки символьних рядків, введення/виведення тощо. Стандартні функції спрощують роботу програміста.

Програміст може написати свої функції, тобто виділити в окремі модулі деякі задачі, до виконання яких можна звертатися з багатьох точок програми. Ці функції найчастіше **називають функціями, що визначаються програмістом.**

Звертання до функції називають **викликом функції**. У виклику функції вказується її ім'я та передається інформація (як аргументи), яка необхідна для виконання функції.

Після виконання функції програма повертається в те місце, звідки відбувався виклик функції. Виклик функції може бути записаний як у головній функції (main), так і в будь-якій іншій функції.

# Схема взаємодії функцій в програмі мовою С



Функції дозволяють розбити програму на модулі.

Усі змінні, які об'явлені в тілі функції, є **локальними змінними** – вони відомі тільки функції, в якій вони визначені.

Більшість функцій мають **список параметрів**. Параметри дозволяють функціям обмінюватися інформацією. Параметри функції – це також локальні змінні.

# Визначення функції

тип поверненого значення ім'я функції  
(список параметрів)

{

об'яви

оператори

}

# Приклади об'яви функцій

```
int Fun1 (int a, int b, int c);
```

```
char Fun2 (int x, int y);
```

```
void Fun3 (char ch, int num, int line);
```

```
float Fun4 (float q, float t, float r, int k) ;
```

```
int Fun5 (void);
```

```
void Fun6 (void);
```

```
float Fun7 (float q, double t, float r, int k) ;
```

Як ім'я функції може бути будь-який допустимий ідентифікатор. Типом результату, який повертає функція, є **тип поверненого значення**. Якщо як тип задано ключове слово `void`, це означає, що функція не повертає нічого. Якщо **тип поверненого значення** не вказаний, компілятор вважає, що тип має значення `int`.



**Список параметрів** - це список об'яв параметрів (відокремлених комами), які отримує функція в момент її виклику. Якщо функція не отримує значень, **список параметрів** позначається ключовим словом **void**. Тип кожного параметра повинен бути описаний, за виключенням типу **int**. Якщо тип не вказаний, вважається, що параметр має тип **int**.

- **Об'яви** та **оператори** у середині фігурних дужок складають ***тіло функції***.

- Перед першим викликом функція повинна бути визначена так, як наведено вище, або за допомогою ***прототипу***.
- Компілятор використовує прототип функції для перевірки того, що виклик функції має коректний тип поверненого значення, коректне число аргументів, коректний тип аргументів і коректний порядок слідування аргументів.

Існують такі способи повернення управління у ту точку програми, в якій була викликана функція.

Якщо функція не повертає результат, управління повертається, як тільки зустрічається права фігурна дужка, що завершує тіло функції,

або при виконанні оператора

***return;***

Якщо функція повертає результат, оператор

***return вираз;*** // ***return( вираз);***

повертає значення ***виразу***.

# Приклад 1

Знайти максимальне з трьох чисел і повернути це значення.

```
#include <stdio.h>
```

```
int maximum(int x, int y, int z);
```

```
main ()
```

```
{  
int a, b, c;  
printf ("Ввести три числа:");  
scanf("%d %d %d", &a, &b, &c);  
printf("Максимальне значення: %d\n", maximum(a, b, c));  
return 0;  
}
```

**/\*Визначення функції maximum\*/**

**int maximum(int x, int y, int z)**

**{ int max=x;**

**if (y > max)**

**max=y;**

**if (z > max)**

**max=z;**

**return max;**

**}**

```
Input 3 numbers:4 4 2  
Max number: 4  
-
```

```
Input 3 numbers:3 -5 7  
Max number: 7  
-
```

```
Input 3 numbers:-9 -3 -5  
Max number: -3  
-
```



# Структура програми включає наступні секції:

preprocessor directives

global declarations

function prototypes

main()

{

local variables to function main ;

statements associated with function main ;

}

int f1(void){

local variables to function 1 ;

statements associated with function 1 ;}

double f2(parametrs){

local variables to function f2 ;

statements associated with function 2 ; }